

## REMARKS/ARGUMENTS

Applicants amended claim 16 to correct a spelling mistake.

### 1. The Claims Comply With the Written Description Requirement

The Examiner rejected claims 1-24 as failing to comply with the written description requirement (35 U.S.C. §112, par. 1) on the grounds that the language in the independent claims of “storing a string of non-Unicode characters in the constant” is not described on pgs. 16, 20, and 21 of the Specification. (Fifth Office Action, pgs. 2-3) Applicants traverse.

Pages 16, 17, and 20 of the Specification show that a define constant “DC” instruction instructs the assembler to convert the characters in the operand field to the proper machine language representation. The characters to convert are in the DC constant. See, pgs. 16, 17, and 20. Applicants submit that this string of characters to convert, such as non-unicode characters, are stored “in the constant”, or in an addressable area addressed by the constant.

Applicants further note that the language to which the Examiner objects was included in original claim 1 of the filed Specification. According to the Manual of Patent Examination and Procedure (“MPEP”), “claims constitute their own description” and that “rejection of an original claim for lack of written description should be rare.” MPEP Sec. 2163.03, pgs. 172 and 186 (Aug. 2005, 8<sup>th</sup> ed., Rev. 3).

For these reasons, Applicants submit that the Specification provides sufficient written description for the “storing a string” limitation found in the original claims.

### 2. Claims 1-24 are Patentable Over the Cited Art

The Examiner rejected claims 1-24 as obvious over the “Description of the Related Art” of the Application (“Related Art”) in view of Lemay (“Laura Lemay’s Web Workshop ActiveX, and VBScript”....), and further in view of Edberg (U.S. Patent No. 5,793,381).

Claims 1, 9, and 17 concern creating a string of Unicode characters stored in a memory of a computer, and require: creating a constant whose data type is a non-Unicode data type, wherein the constant specifies non-Unicode data to convert to Unicode; storing a string of non-Unicode characters in the constant which is stored in the memory of the computer; retrieving a

specification of a code page in which the non-Unicode character string is encoded; translating the non-Unicode character string stored with the constant into a Unicode character string responsive to the specification of the code page; and storing the Unicode character string in the constant stored in the memory of the computer.

The Examiner cited pg. 5, lines 10-28 of the Related Art of the Application (“Related Art”) as teaching the claim requirement of storing a string of non-Unicode characters with a constant whose data type is Unicode, which is stored in the memory of the computer. (Final Office Action, pg. 4) Applicants traverse.

The cited pg. 5 mentions different representations of a character string in different character formats and shows how after Unicode translation, the same characters are represented by twelve bytes. Although the cited pg. 5 discusses Unicode translation, nowhere does this cited pg. 5 anywhere disclose or mention creating a constant, storing a string of non-Unicode characters in the constant, that specifies a non-Unicode character string to convert to Unicode. There is no teaching or mention in the cited pg. 5 of creating a constant to store the non-Unicode character being translated. Moreover, there is no teaching in the cited pg. 5 of a constant that specifies non-Unicode data to convert to Unicode. Instead, the cited pg. 5 discusses converting a character from one format, such as hexadecimal, to Unicode. There is no teaching of the claimed use of a constant to accomplish the translation.

The Examiner then cited p. 75, paras. 4-7 and p. 78 of Lemay as teaching creating and declaring a constant data type to which values are assigned. (Fifth Office Action, pg. 4) Applicants submit that nowhere does the cited Lemay teach or suggest the claim requirement of a constant specifying non-unicode data to convert to unicode.

The cited pg. 75 mentions that a constant is a variable whose value is fixed throughout its lifetime and that you declare a constant with the CONST directive. If any part of your program attempts to change that value in the constant, a runtime error is generated. The cited pg. 75 further mentions that VBScript does not have a CONST directive , and that the safest way to declare a constant is to use a naming convention.

Nowhere does the cited p. 75 anywhere teach or suggest a constant whose data type is a non-Unicode data type, wherein the constant specifies non-Unicode data to convert to Unicode, such that the value or non-Unicode data in the constant is translated to Unicode. In fact, the cited

pg. 75 teaches away from this requirement that the non-Unicode string in the constant be translated to a different value because the cited pg. 75 mentions that an error is generated if there is an attempt to change the value in the constant. Thus, not only is there no teaching or mention in the cited pg. 75 of a constant specifying a non-Unicode data type to convert to Unicode, but the cited pg. 75 teaches away from changing the value in the constant.

The cited pg. 78 mentions that a value entered into a text box is converted into an integer than assigned to a local variable. The content of the local variable is then multiplied (in Line 11, pg. 74) by a constant value of 3 (MY\_GLOBAL\_CONSTANT = 3 in Line 7) and this new value is placed back in the local variable. This calculated value of the local variable is then shown in the alert box.

Although the cited pg. 78 discusses multiplying the value in a local variable, nowhere does the cited pg. 78 anywhere disclose a constant the specifies to convert a non-Unicode character to a Unicode data type. In other words, even if one were to liken the local variable of Lemay with the claimed constant, as the Examiner appears to do, nowhere does the cited p. 78 anywhere disclose that this local variable specify to convert the content of the local variable to a different data type, i.e., from non-Unicode to Unicode. Instead, the cited pg. 78 discusses a simple multiplication operation on the content of a local variable that does not appear to convert its data type, just change the value.

There is no teaching or suggestion in the cited art of a constant specifying non-Unicode data to convert to Unicode, and then translating the non-Unicode data stored in the constant to Unicode.

Accordingly, amended claims 1, 9, and 17 are patentable over the cited art because the cited art does not teach or suggest all the claim requirements.

Claims 2-8, 10-16, and 18-24 are patentable over the cited art because they depend from one of claims 1, 9, and 17. Moreover, the following dependent claims provide additional grounds of patentability over the cited art.

Claims 5, 13, and 21 depend from claims 1, 9, and 17 and further recite that the translation is performed by the computer according to a scope, wherein the specification of the code page applies to translate constants in a portion of a computer program identified by the scope.

The Examiner cited col. 3, lines 57-61 and col. 4, lines 10-67 of Edberg as teaching the additional requirements of these claims. (Fifth Office Action, pg. 6) Applicants traverse.

The cited col. 3 mentions dividing a source string in a first character encoding into text elements, looking-up in a mapping table a conversion code associated with a second character encoding for each text element, and combining the conversion codes of the text elements to form the target string. Nowhere does this cited col. 3 anywhere teach or suggest the claim requirement of the use of a scope such that the specification of the code page applies to translate constants in a portion of a computer program identified by the scope.

The cited col. 4 also mentions looking up target encodings in a mapping table for source target encodings. The code conversion system may include a fallback handler and a scanner table, such that the fallback handler provides codes when the lookup handler is unable to provide a code for one or more text elements, where the fallback handler codes are not exactly equivalent to the source text elements, but similar in appearance. The cited col. 4 further discussing scanning an input character having a character encoding, and each character of the input character string having a character class. The scanning system determines whether the input character of the input character string should be included within a current text element or whether the current text element should end a new text element begun.

Although the cited col. 4 discusses converting a source string to a conversion code by looking up conversion codes in a table or mapping, nowhere does the cited col. 4 anywhere teach or suggest the claim requirement of the use of a scope such that the specification of the code page applies to translate constants in a portion of a computer program identified by the scope. Instead, the cited cols. 3-4 discuss a mapping table that stores target codings for the conversions. However, there is no teaching or suggestion of a scope for these mappings and conversion codes, such that they apply to a portion of the computer program identified by the scope.

Accordingly, amended claims 5, 13, and 21 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Amended claims 6, 14, and 22 depend from claims 5, 13, and 21 and further require that the scope is global, the global scope specifying that the specification of the code page applies to translate constants in the entire computer program.

The Examiner cited pg. 75, paras. 8-77 of Lemay as teaching these claim requirements.  
(Fifth Office Action, pg. 7)

The cited pg. 75 of Lemay discusses the scope of a constant that indicates the extent to which procedures and subroutines can use a constant. Although the cited pg. 75 discusses the scope of what procedures may use a constant, nowhere does the cited pg. 75 anywhere teach or suggest the claim requirement of a global scope that specifies that the specification of the code page that applies to translate constants in the entire computer program. There is no mention in the cited Lemay of a scope or other value indicating that the code page is supposed to apply to translate non-Unicode data in all constants in the computer program. Instead, Lemay discusses the scope of procedures and subroutines that may use the constant.

Accordingly, amended claims 6, 14, and 22 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 7, 15, and 23 depend from claims 5, 13, and 21 and further require that the scope is local, the local scope specifying that the specification of the code page applies to translate constants in a subsequent portion of the computer program.

The Examiner cited the above discussed pg. 75 of Lemay as teaching these claim requirements. (Fifth Office Action, pg. 7) Applicants traverse.

The above discussed cited pg. 75 of Lemay discusses the scope of a constant that indicates the extent to which procedures and subroutines can use a constant. Although the cited pg. 75 discusses the scope of what procedures may use a constant, nowhere does the cited pg. 75 anywhere teach or suggest the claim requirement of a local scope that specifies that the specification of the code page applies to translate constants in a subsequent portion of the computer program. There is no mention in the cited Lemay of a scope or other value indicating that the code page is supposed to apply to translate non-Unicode data in all constants in a subsequent portion of the computer program.

Accordingly, claims 7, 15, and 23 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

Claims 8, 16, and 24 depend from claims 5, 13, and 21 and further require that the scope is constant specific and that the constant specific scope specifies that the code page applies only to a specific constant.

The Examiner cited pg. 5, lines 10-28 of the Related Art with respect to these claims.  
(Fifth Office Action, pg. 8) Applicants traverse.

The cited pg. 5 discusses converting a character string in one format, such as hexadecimal, to a Unicode format. Nowhere does the cited pg. 5 anywhere teach or suggest a constant specific scope that specifies that the code page applies only to a specific constant. Instead, the cited pg. 5 discusses translation of a character string.

Accordingly, claims 8, 16, and 24 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not taught or suggested in the cited art.

3. Added Claims 25-27 are Patentable Over the Cited Art

Added claims 25-27 depend from claims 1, 9, and 17 and further require that the constant is of a named type that specifies that the constant content is to be converted to Unicode.

The added requirements of these claims are disclosed on at least pgs. 16-22, which discloses a “DC” define constant that directs the assembler to convert the characters in the defined constant. These pages discuss named constants “DC C”, “DC G”, “DC CU”, “DC GU”, etc., specifying a particular conversion to be performed.

Applicants submit that these added claims are patentable over the cited art because they depend from base claims 1, 9, and 17, which are patentable over the cited art for the reasons discussed above and because the additional requirements of these claims in combination with the base claims provide further grounds of patentability over the cited art.

Conclusion

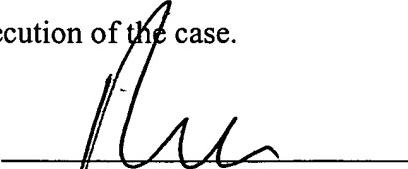
For the above reasons, Applicant submits that the pending claims 1-27 are in condition for allowance. Applicants submit herewith the fee for the added claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

Amdt. dated April 10, 2006  
Reply to Office action of Jan. 10, 2006

Serial No. 09/613,083  
Docket No. STL920000055  
Firm No. 0054.0038

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: April 10, 2006

By: 

David W. Victor  
Registration No. 39,867

Please direct all correspondences to:

David Victor  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: 310-553-7977  
Fax: 310-556-7984